**校园社区平台**

作者姓名：赵安玉

专业名称：软件工程

指导教师：张玺 助教

**成都理工大学工程技术学院**

**学位论文诚信承诺书**

**本人慎重承诺和声明：**

1.本人已认真学习《学位论文作假行为处理办法》（中华人民共和国教育部第34号令）、《成都理工大学工程技术学院学位论文作假行为处理实施细则（试行）》（成理工教发〔2013〕30号）文件并已知晓教育部、学院对论文作假行为处理的有关规定，知晓论文作假可能导致作假者被取消学位申请资格、注销学位证书、开除学籍甚至被追究法律责任等后果。

2.本人已认真学习《成都理工大学工程技术学院毕业设计指导手册》，已知晓学院对论文撰写的内容和格式要求。

3.本人所提交的学位论文（题目：网上拍卖系统），是在指导教师指导下独立完成，本人对该论文的真实性、原创性负责。若论文按有关程序调查后被认定存在作假行为，本人自行承担相应的后果。

承诺人（学生签名）：

20 年 月 日

注：学位论文指向我校申请学士学位所提交的本科学生毕业

**摘要**

**关键词**：校园社区平台 SpringBoot GraphQL JavaWeb Vue

**Abstract**

目录

[摘要 I](#_Toc846930886)

[Abstract 2](#_Toc1681692777)

[前言 - 1 -](#_Toc1714636915)

[1 绪论 - 2 -](#_Toc1957747793)

[2 校园社区平台技术概述 - 2 -](#_Toc424238335)

[2.1 概念 - 2 -](#_Toc719885386)

[2.1.1 基本概念 - 2 -](#_Toc1649760492)

[3 平台架构设计 - 3 -](#_Toc596516649)

[3.1 开发技术及版本依赖 - 3 -](#_Toc1189641421)

[3.1.1 开发环境 - 3 -](#_Toc1025202362)

[3.1.2 开发技术 - 3 -](#_Toc1350490027)

[3.2 平台架构 - 3 -](#_Toc783368690)

[3.2.1 平台后端模块概述 - 3 -](#_Toc1102520059)

[3.2.2 平台前端概述 - 4 -](#_Toc2044897763)

[3.2.3 平台前后端通信技术 - 4 -](#_Toc1967513926)

[4 平台详细设计及实现 - 5 -](#_Toc1365180540)

[4.1 平台数据存储设计 - 5 -](#_Toc1540383426)

[4.1.1 MySQL数据库表设计 - 5 -](#_Toc304089172)

[4.2 平台后端设计与实现 - 8 -](#_Toc1303455736)

[4.2.1 功能模块设计及概要 - 8 -](#_Toc35005211)

[4.2.2 部分功能设计与实现 - 18 -](#_Toc521595368)

[4.3 平台前端设计与实现 - 18 -](#_Toc294702567)

[4.3.1 前端技术选取分析 - 18 -](#_Toc1726956429)

[4.3.2 页面设计与布局 - 18 -](#_Toc336465782)

[4.4 平台前后数据交互策略及实现 - 18 -](#_Toc861021530)

[4.4.1 交互技术选取技术分析 - 18 -](#_Toc278722862)

[4.4.2 数据通信详细设计 - 18 -](#_Toc233665123)

[4.5.3 数据交互详细实现 - 18 -](#_Toc2145174067)

[5 测试与分析 - 19 -](#_Toc468703135)

[5.1 测工具介绍 - 19 -](#_Toc1101513929)

[5.2 测试技术与方法概述 - 19 -](#_Toc1801979802)

[5.2 测试用例 - 19 -](#_Toc1315634022)

[总结 - 20 -](#_Toc635723058)

[参考文献 - 21 -](#_Toc1369133069)

**前言**

**1 绪论**

# 2 校园社区平台技术概述

**2.1 概念**

### 2.1.1 基本概念

1. J2EE的概念

J2EE安全技术是一种通过使用了Java的平台和数据库来有效简化与企业和公司系统解决方案的设计和开发,增长和有效管理与业务有关的复杂安全问题的系统结构。

1. 模块化设计的概念和思想

模块化[程序设计](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1/223952" \t "/home/anyu/Documents\\x/_blank)是指在进行[程序设计](https://baike.baidu.com/item/%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1" \t "/home/anyu/Documents\x/_blank)时将一个大程序按照功能划分为若干小程序模块，每个小程序模块完成一个确定的功能，并在这些模块之间建立必要的联系，通过模块的互相协作完成整个功能的程序设计方法。在设计较复杂的程序时，一般采用自顶向下的方法，将问题划分为几个部分，各个部分再进行细化，直到分解为较好解决问题为止。模块化的目的是为了降低程序复杂度，使程序设计、调试和维护等操作简单化。

1. 前后端分离架构技术概念
2. MVC 设计模式概念
3. Spring 框架

Spring框架是[Java平台](https://baike.baidu.com/item/Java%E5%B9%B3%E5%8F%B0/3793459" \t "/home/anyu/Documents\\x/_blank)上的一种开源应用框架，提供具有控制反转特性的容器。尽管Spring框架自身对编程模型没有限制，但其在Java应用中的频繁使用让它备受青睐，以至于后来让它作为[EJB](https://baike.baidu.com/item/EJB/144195" \t "/home/anyu/Documents\\x/_blank)（EnterpriseJavaBeans）模型的补充，甚至是替补。它简化了 Java 企业级应用开发。

1. SpingBoot 概念
2. MybatisPlus 框架概念
3. GraphQL API 查询语言的概念
4. JWT (Java Web Token)的概念

# 3 平台架构设计

**3.1 开发技术及版本依赖**

本节主主要介绍校园社区平台所用到的技术和开发环境，以及所依赖其他资源。

### 3.1.1 开发环境

操作系统：Linux 510

开发工具：IDEA 2020.3 Navicat

数据库：MySQL 8.0.20 Redis

容器技术：Docker

测试工具：Postman

SpringBoot 2.3.4.RELEASE

MybatisPlus 3.4.0

Java 11

Gradle 6.7

Vue 2

### 3.1.2 开发技术

**3.2 平台架构**

平台是一个前后端分离的web应用。后端使用了Gradle作为版本控制工具和模块管理工具；基于SpringBoot 的框架做快速开发; MybatisPlus做持久层框架,进行数据库操作。前端使用Vue框架搭建前端应用，利用Element 进行快速页面搭建。使用GraphQL作为前后端通信方式。

### 3.2.1 平台后端模块概述

各个模块遵循软件开发模块化设计原则，每个模块都只关注自己的功能或业务。其中ancf-msg、ancf-user、ancf-post业务模块按MVC模式实现，它们是和前端进行交互的模块；理论上 ancf-auth、ancf-oss、ancf-cache对外部模块只暴露业务接口，它们只负责对内部模块提供服务。

1、 ancf-webapp 模块，它是程序的主入口模块，最顶层模块；直接依赖于ancf-service-msg、ancf-service-user、ancf-service-post业务模块。程序的配置文件存放该模块中。

2、ancf-service-msg 模块，它是校园社区平台用户间消息发送、平台消息推送的业务实现。

3、 ancf-service-user 模块，它是校园社区平台与用户相关业务的实现，包括用户注册、用户登录、头像修改等等。

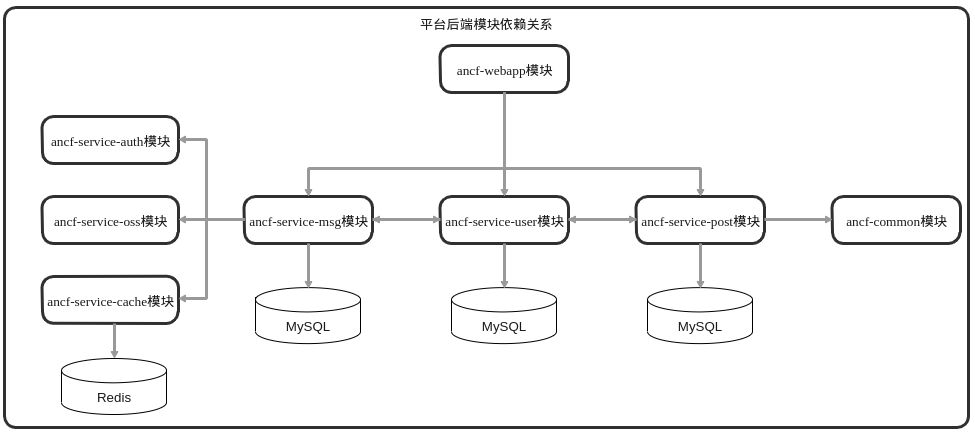
4、 ancf-service-post 模块，它是校园社区论坛与帖子相关业务的实现，包括发帖，评论、点赞等等。

5、ancf-common 模块，它是公共模块，最底层模块，基本上所有的模块都依赖于它。

6、ancf-service-auth 模块，它是平台权限认证、安全管理的实现。

7、 ancf-service-oss模块，它是平台文件上传云端服务器的实现，借助于阿里云OSS服务。

8、 ancf-sesrvice-cache 模块，它是系统缓存的业务实现。



### 3.2.2 平台前端概述

### 3.2.3 平台前后端通信技术

# 4 平台详细设计及实现

**4.1 平台数据存储设计**

关系型数据库使用MySQL，它作为平台主要的数据存储数据库，小部分实时热点数据或者有效期的数据使用Redis作为缓存数据库。

### 4.1.1 MySQL数据库表设计

1. 用户表 user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | ID |
| account | char | 16 | NOT NULL | 账号 |
| gender | tinyint | 1 | NOT NULL | 性别 |
| age | tinyint | 3 | NOT NULL | 年龄 |
| birthday | datetime | - | NOT NULL | 出生日期 |
| nickname | char | 9 | NOT NULL | 昵称 |
| real\_name | char | 6 | NULL | 真实姓名 |
| avatar | char | 100 | NOT NULL | 头像存储地址 |
| email | char | 25 | NULL | 邮箱地址 |
| mobile | char | 11 | NULL | 手机号 |
| salt | char | 5 | NOT NULL | 密码加盐 |
| password | char | 50 | NOT NULL | 密码 |
| activation | tinyint | 1 | NOT NULL | 账号是否激活 |
| status | tinyint | 1 | NOT NULL | 用户状态 |
| create\_time | datetime | - | NOT NULL | 用户创建时间 |
| modified\_time | datetime | - | NOT NULL | 用户修改时间 |

1. 评论表 comment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | 评论id |
| content | varchar | 255 | NOT NULL | 内容 |
| user\_id | int | 10 | NOT NULL | 评论者id |
| entity\_id | int | 10 | NOT NULL | 评论的实体id |
| target\_id | int | 10 | NULL | 被评论者的id |
| entity\_type | tinyint | 1 | NOT NULL | 评论实体类型 |
| status | tinyint | 1 | NOT NULL | 评论状态 |
| create\_time | datetime | - | NOT NULL | 评论创建时间 |
| modifid\_time | datetime | - | NOT NULL | 评论修改时间 |

1. 帖子表 post

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | 帖子id |
| type | tinyint | 1 | NOT NULL | 帖子类型 |
| user\_id | int | 10 | NOT NULL | 发帖者id |
| title | varchar | 20 | NOT NULL | 帖子标题 |
| content | text | - | NOT NULL | 内容 |
| cmt\_num | int | 10 | NOT NULL | 评论数 |
| score | float | - | NOT NULL | 得分 |
| status | tinyint | 1 | NOT NULL | 帖子状态 |
| create\_time | datetime | - | NOT NULL | 创建时间 |
| modified\_time | datetime | - | NOT NULL | 修改时间 |

1. 消息表 message

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否 NULL | 注释 |
| id | int | 10 | NOT NULL | 消息id |
| from\_id | int | 10 | NOT NULL | 发送者id |
| to\_id | int | 10 | NOT NULL | 接受者id |
| chart\_id | varchar | 20 | NOT NULL | 聊天id |
| content | varchar | 255 | NOT NULL | 内容 |
| status | tinyint | 1 | NOT NULL | 状态 |
| create\_time | datetime | - | NOT NULL | 创建时间 |
| modified\_time | datetime | - | NOT NULL | 修改时间 |

**4.1.2 Redis缓存设计**

1. 设计缘由以及问题分析

考虑到部分数据修改和查询频繁，MySQL数据库压力过大，和程序效率问题

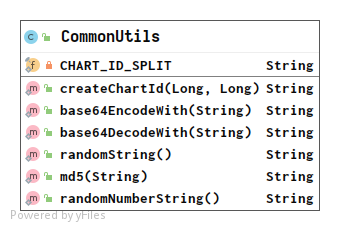
一部分数据具有时效性，过期的数据没有任何可利用价值。

**4.2 平台后端设计与实现**

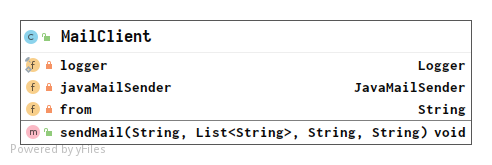
### 4.2.1 功能模块设计及概要

**1、ancf-common**

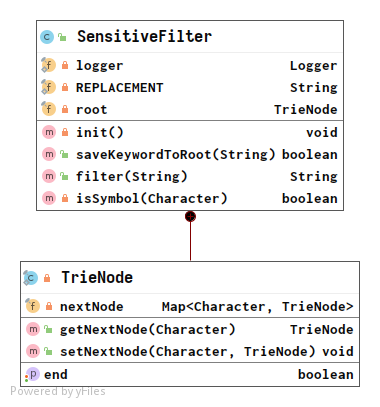
1. CommonUtils 平台公共工具类

****

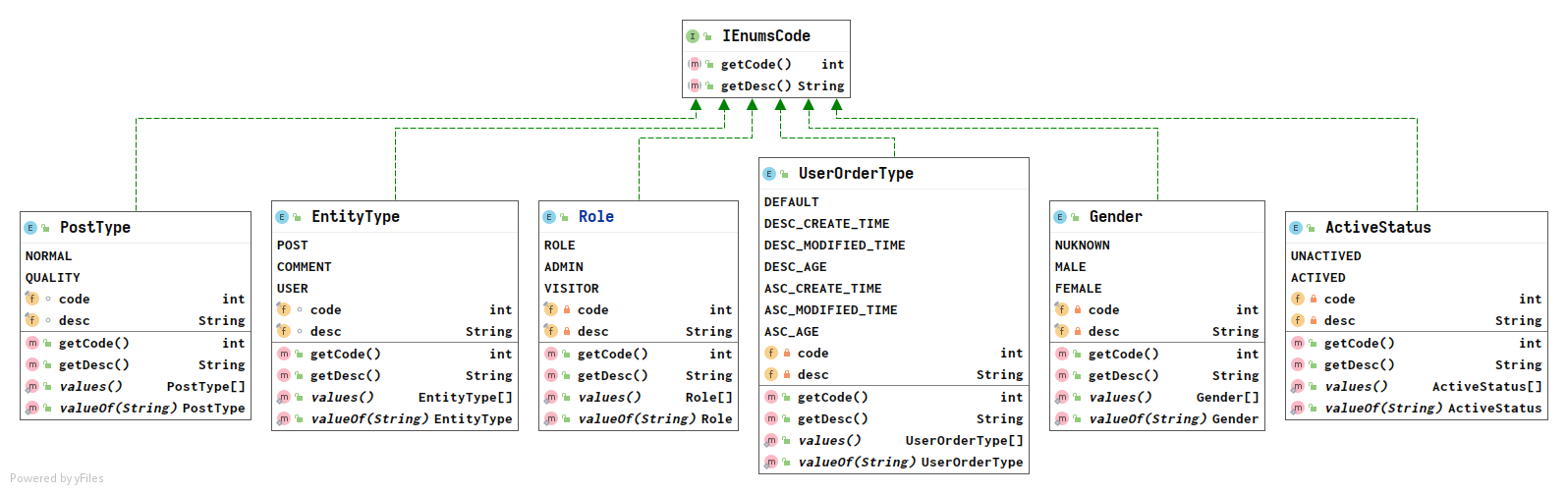
1. MailClient 系统邮件发送组件类

****

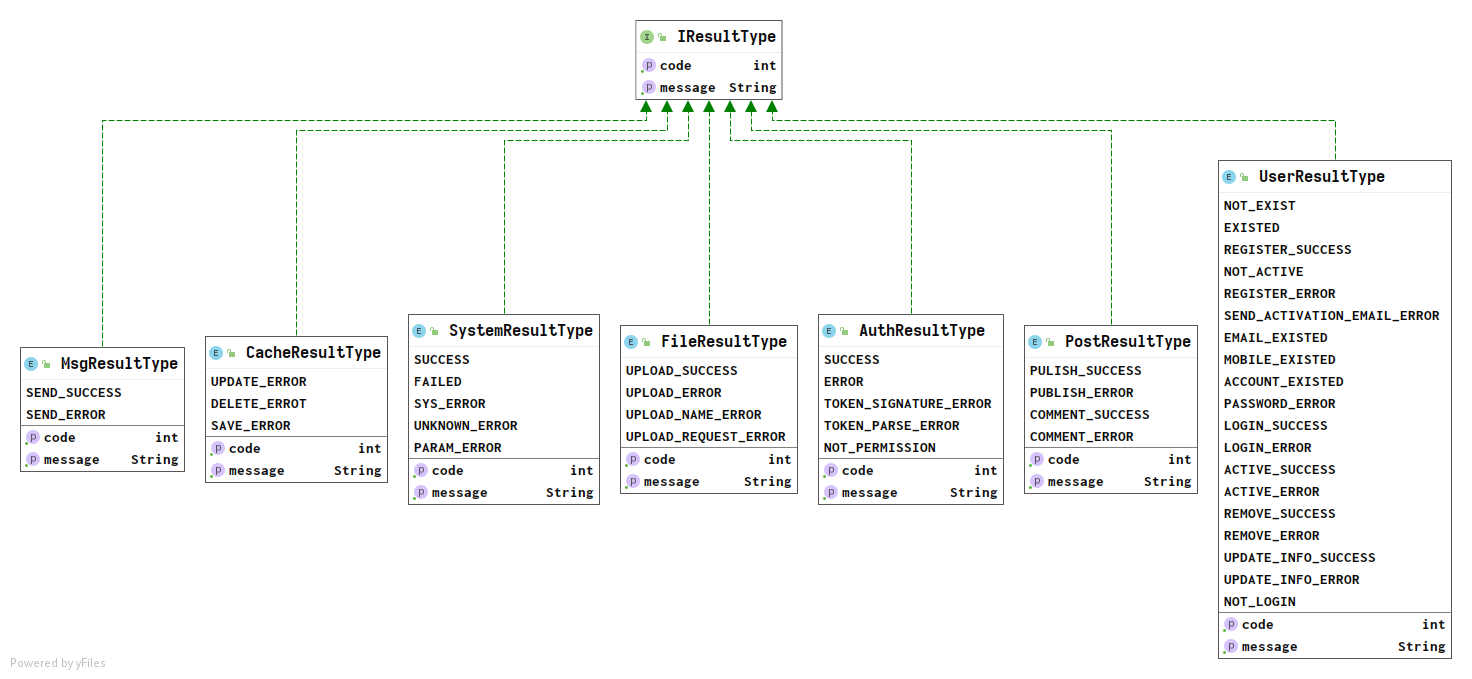
1. 平台敏感字文本文件过滤类

****

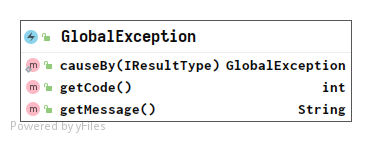
1. 数据库字典枚举类实现

****

1. 业务处理结果枚举类

****

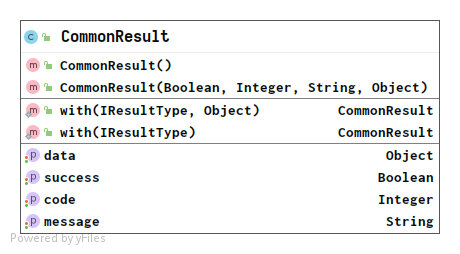
1. 自定义全局异常



1. 全局异常处理

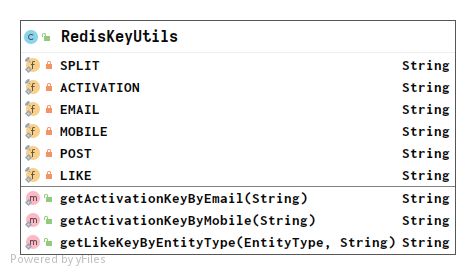


1. 统一结果返回

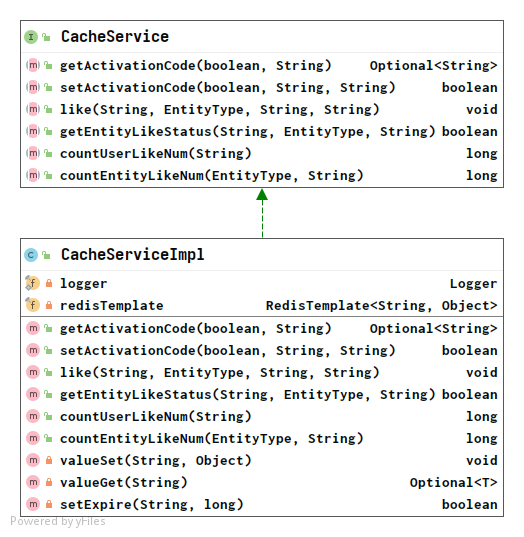


2、ancf-service-cache模块

1. Redis 键生成工具类



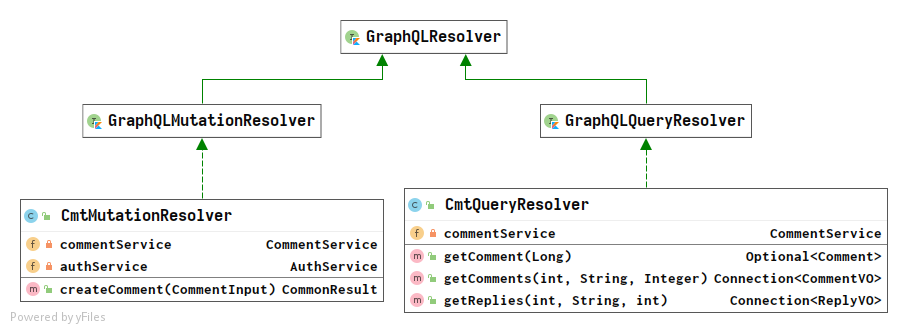
1. 缓存服务接口及实现

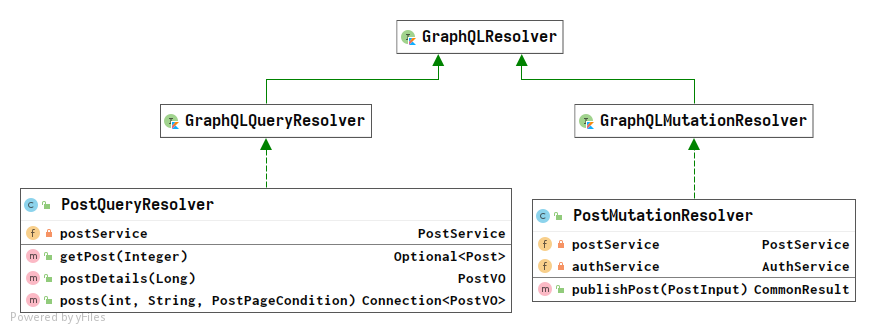
****

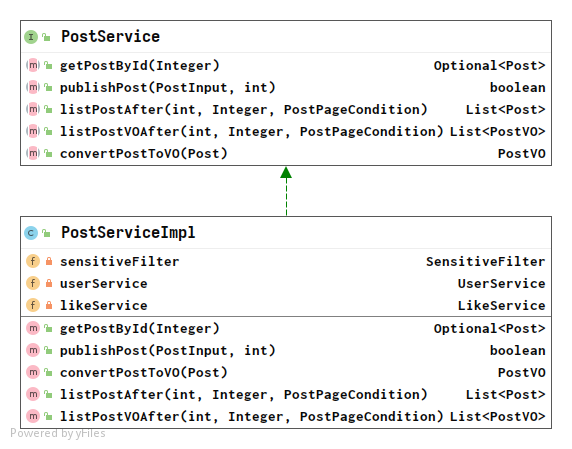
3、ancf-service-msg 模块

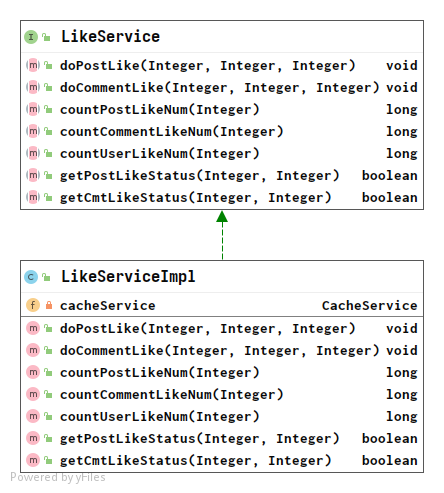
4、ancf-service-post 模块

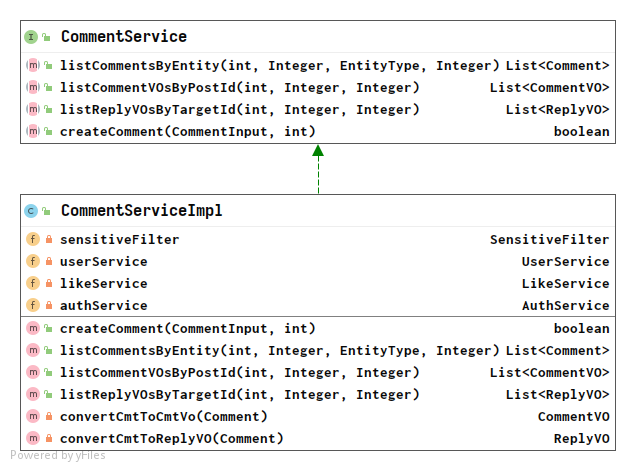
1.

****

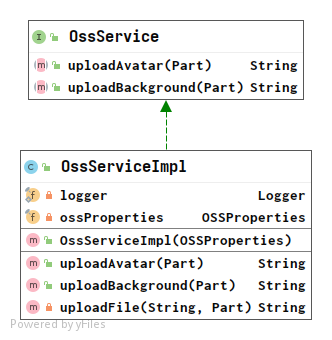
****

****

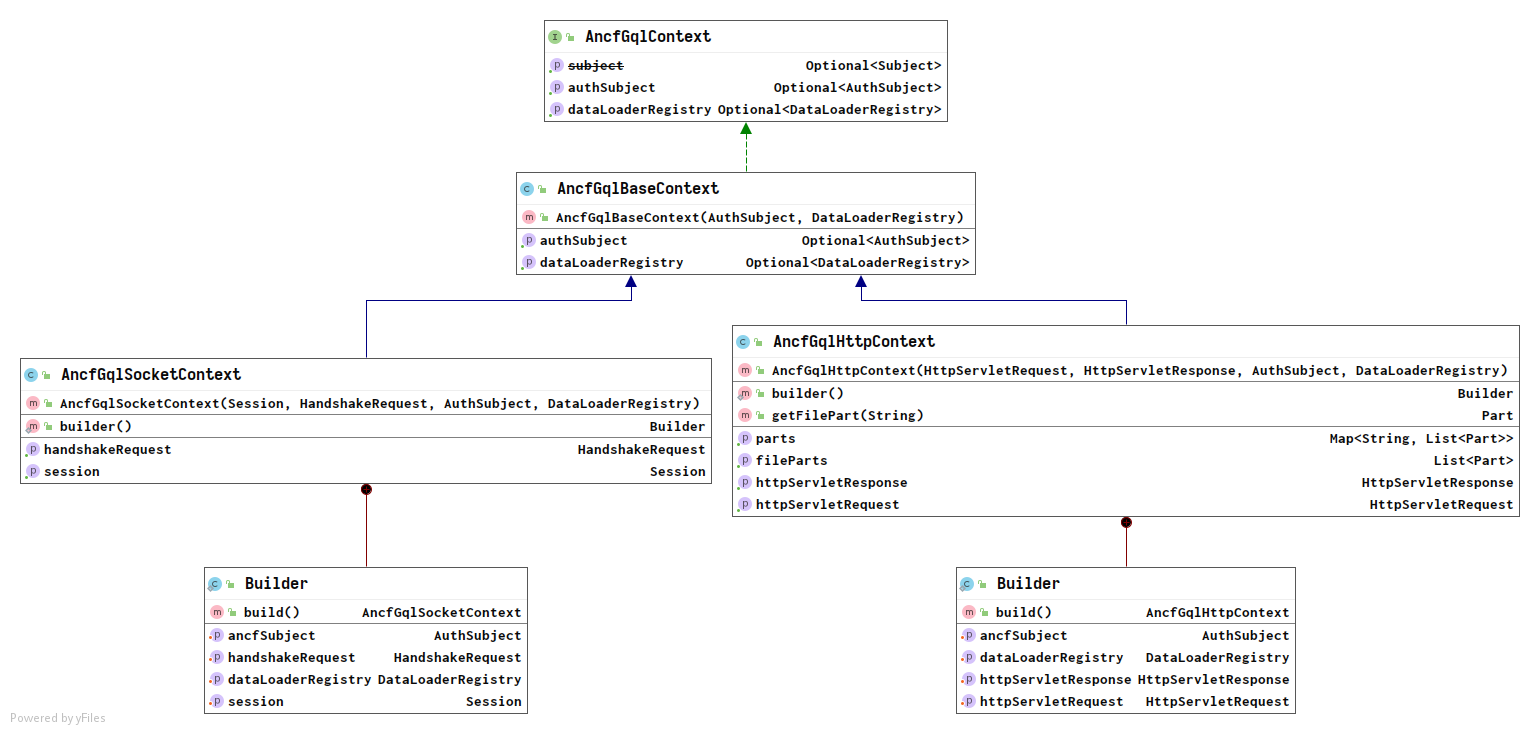
****

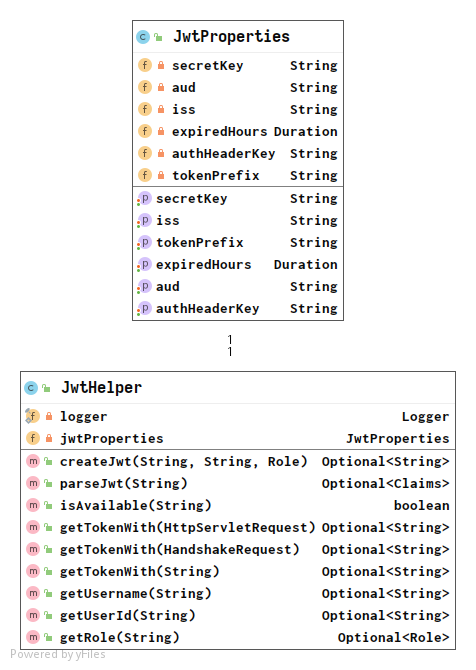
****

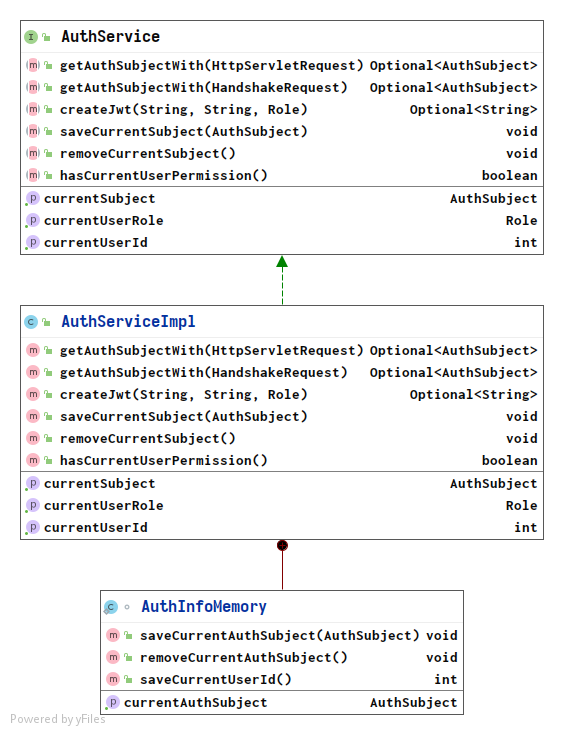
5、ancf-service-oss模块



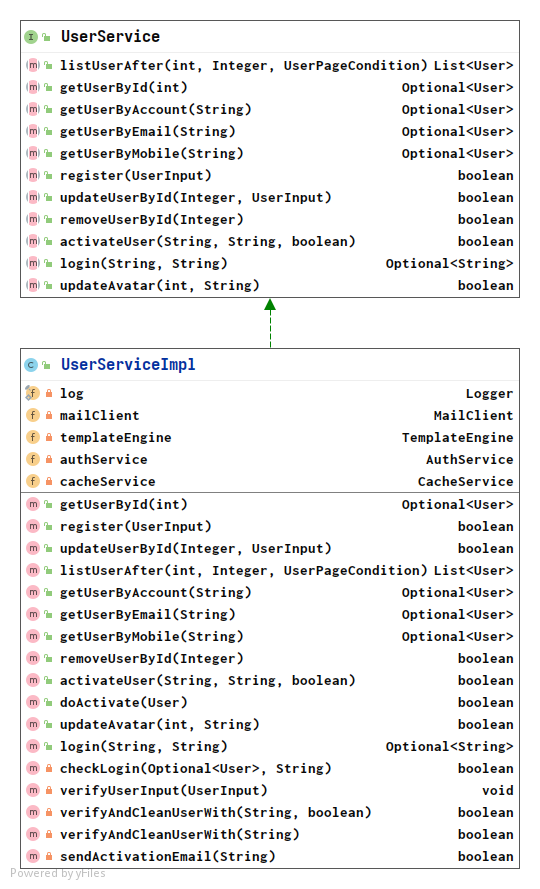
6、ancf-service-auth模块

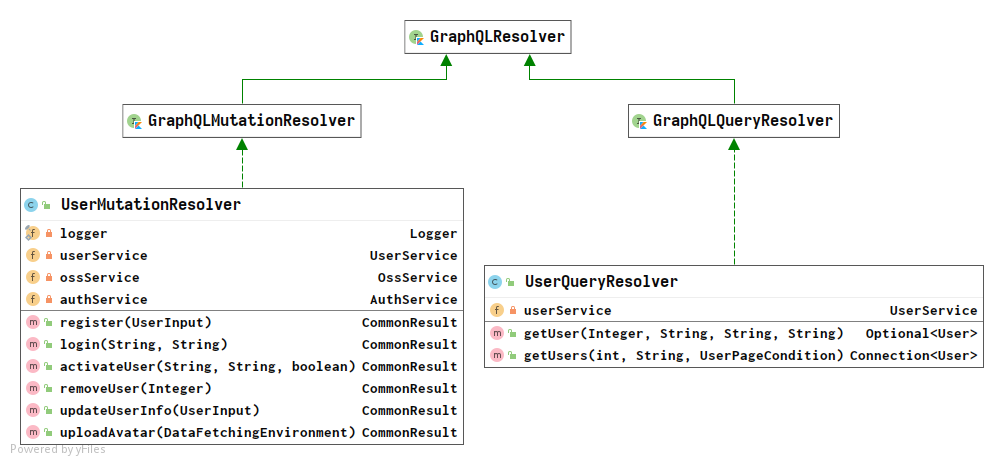
****

****

****

**7、**ancf-service-user模块

****

****

**8、**ancf-webapp模块

### 4.2.2 部分功能设计与实现

**4.3 平台前端设计与实现**

### 4.3.1 前端技术选取分析

### 4.3.2 页面设计与布局

**4.4 平台前后数据交互策略及实现**

### 4.4.1 交互技术选取技术分析

### 4.4.2 数据通信详细设计

### 4.5.3 数据交互详细实现

# 5 测试与分析

**5.1 测工具介绍**

**5.2 测试技术与方法概述**

**5.2 测试用例**

**总结**

**参考文献**